NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

EXPOSING VITAL FORENSIC ARTIFACTS OF USB DEVICES IN THE WINDOWS 10 REGISTRY

by

Jason S. Shaver

June 2015

Thesis Advisor:	Neil Rowe
Second Reader:	Michael McCarrin

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704–0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** June 2015	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE EXPOSING VITAL FORENSIC ARTIFACTS OF USB DEVICES IN THE WINDOWS 10 REGISTRY	**5. FUNDING NUMBERS**
6. AUTHOR(S) Jason S. Shaver	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number ____N/A____.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited	**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

Digital media devices are regularly seized pursuant to criminal investigations and Microsoft Windows is the most commonly encountered platform on seized computers. Microsoft recently released a technical preview build of their Windows 10 operating system which can run on computers, smart phones, tablets, and embedded devices. This work investigated the forensically valuable areas of the Windows 10 registry. The focus was on the Windows Registry hives affected when USB storage devices are connected to a laptop configured with Windows 10. Paths were identified that indicate the date/time of last insertion and removal of a thumb drive. Live monitoring and post-mortem forensic methodologies were used to map Registry paths containing USB identifiers such as make/model information, serial numbers and GUIDs. These identifiers were located in multiple paths in the allocated and unallocated space of the Registries analyzed.

14. SUBJECT TERMS Windows Registry, computer forensic	**15. NUMBER OF PAGES** 63
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

THIS PAGE INTENTIONALLY LEFT BLANK

EXPOSING VITAL FORENSIC ARTIFACTS OF USB DEVICES IN THE WINDOWS 10 REGISTRY

Jason S. Shaver
Computer Forensic Agent, Homeland Security Investigations
B.A., Towson University, 2001
M.S., University of Phoenix, 2005

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN CYBER SYSTEMS AND OPERATIONS

from the

NAVAL POSTGRADUATE SCHOOL
June 2015

Author:	Jason S. Shaver
Approved by:	Neil Rowe Thesis Advisor
	Michael McCarrin Second Reader
	Cynthia Irvine Chair, Cyber Academic Group

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Digital media devices are regularly seized pursuant to criminal investigations and Microsoft Windows is the most commonly encountered platform on seized computers. Microsoft recently released a technical preview build of their Windows 10 operating system that can run on computers, smart phones, tablets, and embedded devices. This work investigated the forensically valuable areas of the Windows 10 registry. The focus was on the Windows Registry hives affected when USB storage devices are connected to a laptop configured with Windows 10. Paths were identified that indicate the date/time of last insertion and removal of a thumb drive. Live monitoring and post-mortem forensic methodologies were used to map Registry paths containing USB identifiers such as make/model information, serial numbers and GUIDs. These identifiers were located in multiple paths in the allocated and unallocated space of the Registries analyzed.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

CFE	Computer Forensic Examiner
MCW	Master Control Workstation
IACIS	International Association of Computer Investigative Specialists
ISO	International Organization for Standardization
OS	Operating System
GB	Gigabyte
BCFE	Basic Computer Forensic Examiner
DHS	Department of Homeland Security

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

My sincere gratitude goes to Dr. Neil Rowe, Michael McCarrin, Dr. Cynthia Irvine, and the entire Cyber Academic Group at the Naval Postgraduate School for their guidance, encouragement, and unrelenting pressure that made this thesis possible. I would like to thank my colleagues at Homeland Security Investigations for the tremendous support that I received throughout this process. Most of all, to my wife and family: Thank you for your unwavering support and daily motivation through all the late nights and weekends I spent working on this degree.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

The types of crimes most commonly investigated by Computer Forensic Examiners (CFE) include child exploitation, identity theft, homicide, and network intrusion. Statistics from the Internet Crime Complaint Center (IC3) operated by the National White Collar Crime Center (NW3C) indicate computer crime rates have continually risen dramatically in the 21st century (Internet Computer Crime Complaint Center [IC3], 2008). The focus of many investigations includes the need to identify digital footprints available on seized computers that assist in re-creating a crime scene and telling the story of the events that occurred. Part of this discipline includes adopting practices that allow the CFE to identify digital storage devices that were connected to a computer at the focus of an investigation via universal serial bus (USB).

The identification of USB-related footprints related to mounted devices is an invaluable part in the investigation of many categories of computer crime. In child exploitation investigations, it is imperative to determine whether relevant files were transferred to or from any connected devices. The successful identification of this action can result in the addition of a distribution charge and a longer prison sentence. Windows Registry artifacts can also serve to create a list of devices that were mounted to the OS and may assist in identifying evidence items that were not recovered in the course of the enforcement action.

In network-intrusion investigations, timeline analysis may show that a compromise occurred on a certain date. Further investigation of software registry data can identify a USB device that was connected to a system and provide a clue to the origin of malware that was introduced thereafter.

The presence of encryption on the USB device can pose a challenge for computer forensic examinations. However, software registry artifacts may still be used to link USB devices containing contraband files back to a computer used to commit a crime. In cases where the encryption is applied only to specific volumes, directories or files, the registry may be extracted and analyzed for vital data. There are also metadata artifacts that will be discussed in the course of this research paper that may indicate files of interest are present on the USB device.

A. PROBLEM STATEMENT

Windows is the most commonly used operating system (OS) the world over. The Windows Registry is an integral component that contains configuration information and artifacts detailing data useful in an investigation concerning the system hardware, software and associated components (Luttgens & Pepe, 2014). Nuances within the Windows Registry have appeared within each version of the Windows OS released. In 2014, Microsoft released a technical build preview of the Windows 10 OS. The Windows 10 OS will be released on July 29, 2015 and will become the default OS installed on many popular computer brands. Significant numbers of individual users will also elect to upgrade their OS version to Windows 10. Therefore, it will be necessary to understand how data will be stored within this new version of Windows.

USB devices are often critical in investigations. It is vital for CFEs to understand where specific Windows Registry artifacts related to USB devices are located and the significance associated with the embedded metadata. Windows Registry evidence can show that a specific device was connected to a computer, when it was last connected and can provide a better awareness of the scope of a crime. There are no papers to date detailing these specific forensic artifacts in the Windows 10 Registry.

B. RESEARCH METHODOLOGY

Previous descriptions of forensic artifacts for earlier versions of the Windows OS have been published by various researchers and organizations. The artifacts were gathered using a variety of forensic tools. Some tools used to gather Windows Registry information were open source and some were commercial, depending on the preference of the researcher. Four forensic tools were investigated for this thesis; each tool had limitations, so it was useful to compare the results obtained from their use:

- Active monitoring of Windows Registry modifications using the Microsoft SysInternals ProcMon tool.

- Hash analysis using the EnCase tool Windows Registry values that were modified.
- Comparative Windows Registry analysis using RegShot.
- Analysis with the Perl scripts in the RegRipper tool.

Data documenting the common results of the Windows Registry value changes was generated in this research. These results should help CFE's in their future investigations of USB artifacts.

The structure for the results discussed in this thesis is as follows:

- Chapter II reviewed why Windows Registry analysis is valuable to forensic examiners and the data that may indicate a USB device was connected to a computer.
- Chapter III covered previous research of Windows Registries with forensic software suites used in the identification artifacts.
- Chapter IV focused on the methodology employed in the documentation of results for this thesis. Four tools were used to obtain results and validate the artifacts that were observed.
- Chapter V discussed the mapped locations of Registry artifacts pertaining to USB devices in the Windows 10 Registry.
- Chapter VI discussed the significance of these results and recommended areas where further research is recommended.

THIS PAGE INTENTIONALLY LEFT BLANK

II. WINDOWS REGISTRY ARTIFACTS

A. COMMONLY USED OPERATING SYSTEMS

The operating systems (OSs) installed on home and business computers easily fall into two categories: Mac OS and Microsoft Windows. Versions of these OS's have been in use for decades. Figure 1 was obtained from netmarketshare.com and shows that Windows is the leading OS of choice for over 90% of worldwide users (Netmarketshare, 2015).

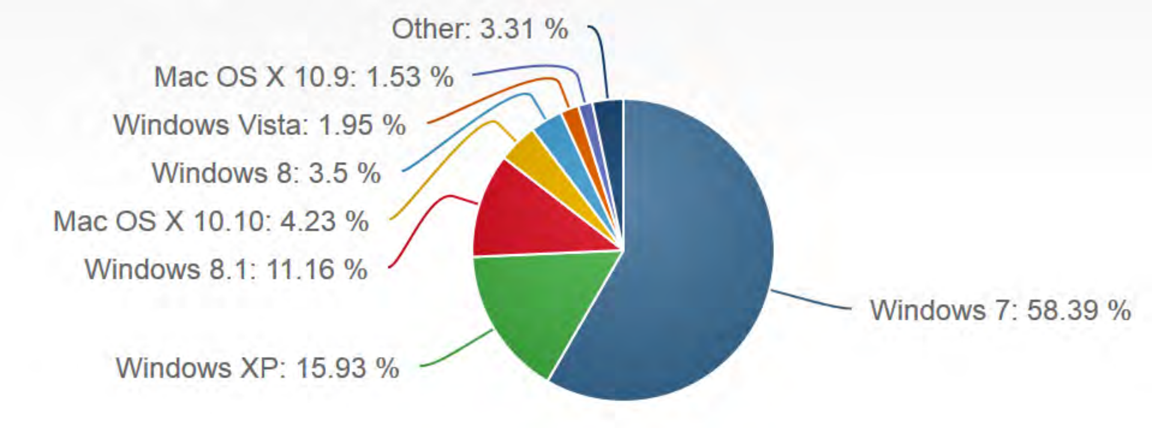


Figure 1. 2015 statistics of desktop computer users (after Netmarketshare, 2015)

The data collected by netmarketshare.com is drawn from a network of over 40,000 websites that span the globe. The values counted by this service are from unique visitors to their network sites.

As a result of this preference, the majority of computers seized pursuant to search warrants by law enforcement entities investigating computer crime will be configured with the Windows OS. It is essential to have a thorough understanding of relevant log and configuration artifacts when examining computers configured with this OS. According to the Figure 1, less than 15% of users have Windows 8 installed, however this is still a higher percentage than users with the Mac OS. Windows 10 users will also make a large part of the population when it is released on July 29, 2015.

B. WHAT IS THE WINDOWS REGISTRY?

At the core of the Windows OS is a collection of forensically useful artifacts that store information vital to the stability and function of the OS: the Windows Registry. The Registry is defined as a central repository or database of the configuration data for the operating system and most of its programs (Bunting, 2012a). The Windows OS uses the Registry to determine permissions assigned to each user, the hardware configured within the computing device and a mapping of the ports that are utilized. The logs and configuration files in the Registry can be significant in a computer forensic investigation to identify activity that took place on a device. The Registry is stored in a proprietary format and only customized tools are useful to access the information stored within.

The Windows Registry is organized in a "hive" structure containing keys, subkeys, values, and supporting files containing backups of significant data. For forensic analysis, the keys and subkeys can be thought of as directories. The values hold data that includes a user's Desktop preferences, time zone information, last shut down date/time, and information pertaining to USB-connected devices. Windows users with "Administrator" accounts can view the Windows Registry hives using the Regedit.exe or Regedt32.exe editing tools installed natively within the OS. Administrators also can modify and backup Windows Registry contents using this tool. An example of the appearance of the Windows Registry while using Regedit is shown in Figure 2.

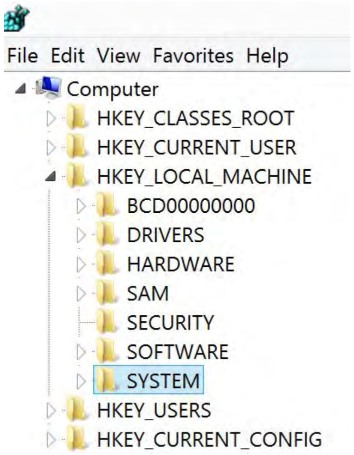


Figure 2. Regedit display of Windows Registry hive

C. LINK FILES

Link files are important to track within the Windows OS. Link files bear the file extension *.lnk* and contain metadata pointers that may be significant in a forensic analysis. Link files are created for a variety of reasons within the Windows OS. They are sometimes specifically created by a user to facilitate access to a file.

Not only can the linked file be within a different directory than the target it points to, it can be stored within a different physical disk and logical volume.

The forensic value of a link is that a link file bearing a pertinent file name may be viewed on a forensic image of an evidentiary item. Even if the file bearing the critical data is not available a link file to it may be present. The metadata within the link file may show that the actual file associated with it was stored on a separate volume. The link files will show the volume that the file of interest was stored on. Our work shows that further information about the missing file can also be discovered.

THIS PAGE INTENTIONALLY LEFT BLANK

III. WINDOWS REGISTRY INFORMATION GATHERING

A. WHY USE A FORENSIC APPROACH?

A forensic approach should be taken to document Registry artifacts. This is important from a scientific perspective and also from an evidentiary perspective. The scientific perspective involves managing a test environment so that quantifiable changes can be attributed to the introduction of a known variable. If this principle is followed, meaningful results can be relied upon. The National Institute of Standards and Technology (NIST) emphasizes the need for forensic results to be repeatable and reproducible (National Institute of Standards and Technology [NIST], 2001). Repeatable conditions are defined as those where independent test results are obtained with the same method on identical test items in the same laboratory by the same operator using the same equipment within short intervals of time (NIST, 2001). Reproducible conditions are those where test results are obtained with the same method on identical test items in different laboratories with different operators using different equipment (NIST, 2001). The results obtained in the course of this research attempt to meet both of the NIST requirements for forensic tool validation.

From the evidentiary perspective, adherence to a sound forensic methodology is necessary for the acceptance of findings to be presented in trial, according to the Federal Rules of Evidence (FRE) 702 Article VII and the Daubert Ruling provide guidance on the competence of an expert witness and the relevance or reliability of the evidence presented (Smith & Bace, 2003). The Daubert Ruling augments the FRE 702 stating that the focus must be solely on the principles and methodology, not on the conclusions they generate. If a validated methodology is used in obtaining a set of results, it will generally meet these evidentiary requirements. These requirements are essential in maintaining the due process rights granted to citizens via the Bill of Rights and the Fifth and Fourteenth Amendments of the Constitution.

The core discipline of computer forensics requires the ability to preserve evidence, identify exactly what occurred on a digital device and the ability to explain why events occurred and why they are relevant to the investigation (IACIS, 2015a). It does not matter whether a forensic examiner is reviewing a device for fraudulently produced identifications or responding to a network compromise. The digital footprints must remain intact so that an investigation can continue and meaningful artifacts can be recovered. Some of the artifacts will include metadata which document dates and times a file was created or modified. The file path where evidentiary files were located may also be significant.

In the course of gathering the research findings for this document, the identification of specific created and modified values within the Registry was crucial. The goal of this work was to map locations where change occurred so that forensic examiners will know that these areas that may contain the crucial "bread crumbs" which show a bigger picture within a Windows 10 computer that they are tasked to investigate.

B. USB ARTIFACTS POINT TO INTERESTING ACTIVITY

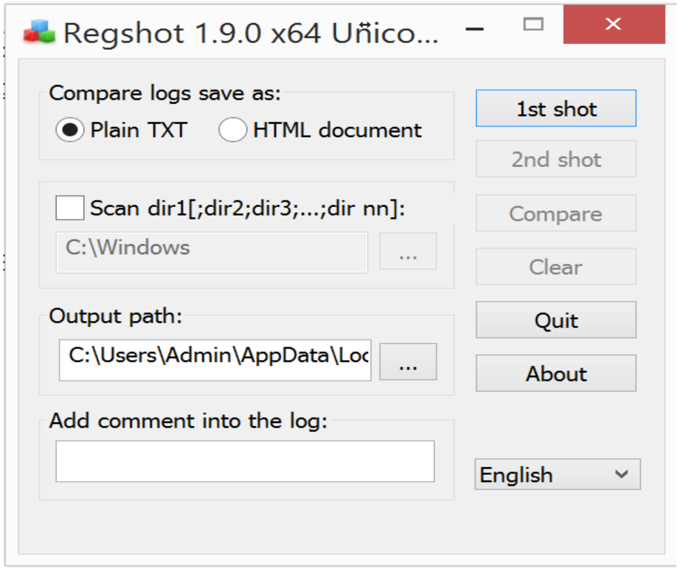
The identification of USB connections can be an important part of a forensic investigation. The action of physically connecting an external storage device requires a purposeful human interaction and indicates a clear level of intent. The action may be completed with the specific intent to install malware, transfer files or to view files stored on a device. Identified activity with an external device can show clear evidence of a user failing to comply with office security policies, or that certain files were knowingly transferred. This thesis focuses on a registry analysis of artifacts related to USB connections.

C. TOOLS OF THE TRADE IN WINDOWS REGISTRY FORENSIC ANALYSIS

Several tools allow users to more easily observe the changes recorded in the Registry as a result of the introduction of an external media device, installation of a program, or other modification to the Windows OS. There are advantages and disadvantages associated with each tool. Rather than rely on a single tool, it is beneficial to use more than one tool when time permits. This practice allows the analyst to validate results, account for a wider scope of forensic artifacts, and maintain proficiency with a variety of forensic software. This affords the forensic examiner the opportunity to fine-tune their tools of choice with the goal of extracting a focused set of accurate data in a time-efficient manner. The sections below discuss some of the tools that were used in previous analyses of mounted-device artifacts in the Windows Registry.

1. RegShot

Regshot is an open source utility that documents and aggregates changes that are made to Windows Registry files. This application is available through the website http://sourceforge.net/projects/regshot/. The program works by taking an initial snapshot of a user-specified directory. After some number of changes, such as the installation of new applications or the introduction of an external digital media device, a second snapshot of the directory is taken and differences are exported into either a plain text or HTML document, depending on user preference. The tool has a simple graphical user interface (GUI) as shown in Figure 3. It is widely used by information technology (IT) specialists, home users and forensic examiners. Monitoring the Registry using RegShot is a convenient way for troubleshooting in the event that installation of a new application has an adverse effect on the operating system.



Compare logs save as:
● Plain TXT ○ HTML document

□ Scan dir1[;dir2;dir3;...;dir nn]:
C:\Windows

Output path:
C:\Users\Admin\AppData\Loc

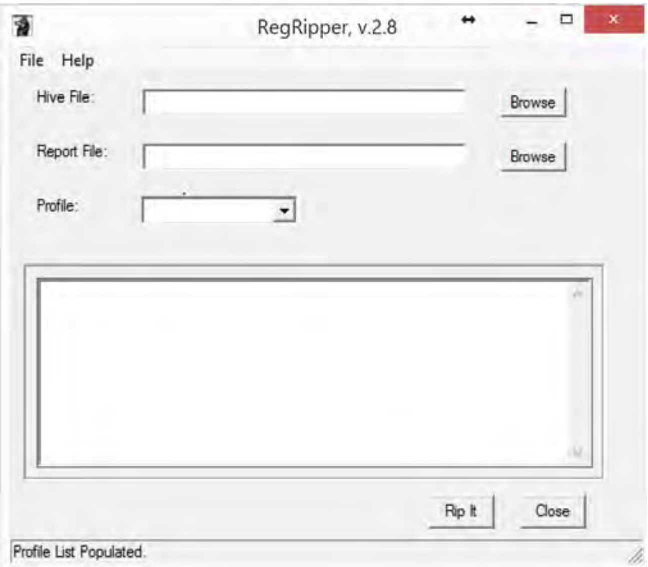
Add comment into the log:

English

Figure 3. Shows a sample display of RegShot v1.9.0

2. RegRipper

Another popular open-source tool for forensic Registry analysis is RegRipper, designed by Harlan Carvey. Figure 4 shows an example of the RegRipper graphical user interface (GUI) in the Windows 8 OS. It is not a comprehensive Registry analysis tool, but a platform for plugins, implemented in Perl, that extract information from registry hives. There are Linux and Windows-based versions of the program along with a set of 324 useful prebuilt plugins that are available for download via https://github.com/keydet89/RegRipper2.8. Eight of these prebuilt plugins focused on mounted device and USB artifact extraction. These plugins can also be custom written by users who possess an understanding of Perl scripting and know the type of details the plugin should parse from the Registry. A plugin that comparatively analyzes Registry changes identified after the introduction of a USB device was not available. A major benefit of this program over RegShot is that a log file is created when RegRipper is run. Log files are beneficial in forensics to add documentation of what processes were run on a specific evidence item. Forensic examiners generally maintain a timeline with detailed notes of what processes are run on evidence items and what the results of the processing were, however the inclusion of a log to accompany the notes can be valuable, especially when testifying in court. Log files automatically document a set of specified activities that occur within a program and their retention is required by the standard operating procedures (SOP) employed by some law enforcement agencies.

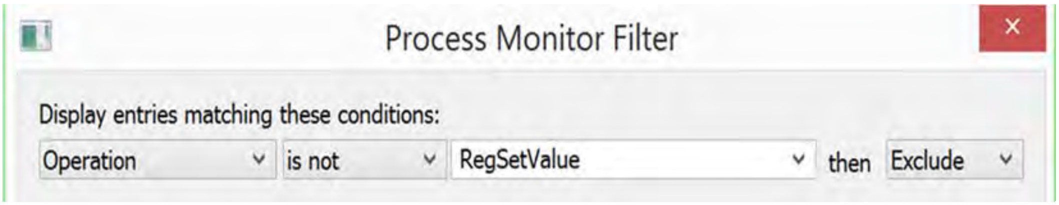


Profile List Populated.

Figure 4. A display of the RegRipper v2.8 program running on the Windows 8 OS.

3. SysInternals Suite

The SysInternals Suite, a free tool by Microsoft, offers a process-monitoring tool known as ProcMon, which allows the user to identify real-time operations on a Windows workstation. The results displayed can be filtered in a number of ways to hone in on a specific data set. An icon on the home screen allows the user to focus the scope of their review to only Registry-related changes. These results can be further filtered by selecting the filter tab and specifying parameters that should exist for items to be displayed or hidden. When using ProcMon to view Registry changes as a result of introducing a new external media device, a filter can be applied to display only write operations to the Registry. This filter can be applied by selecting the options displayed in Figure 5.



Display entries matching these conditions:

Operation is not RegSetValue then Exclude

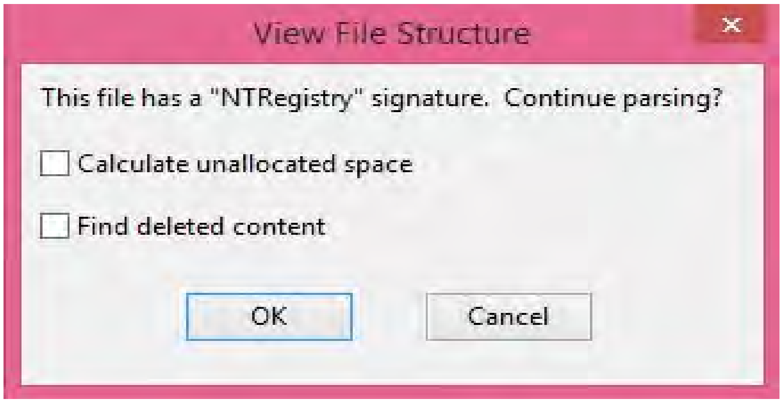
Figure 5. ProcMon filter to display write operations to Registry (after Bunting, 2012b)

After the filter is set and a mounted device is introduced, the resulting changes can be captured. The user can press the "CTRL E" key combination to begin and stop the recording. The feature is useful in narrowing the scope of observed events.

4. EnCase Enterprise

The EnCase platform also provides the forensic examiner with features for documenting changes to the Windows Registry. EnCase is the only commercially licensed software that was used for this thesis. The capabilities of value for this thesis were its ability to analyze records from the allocated and unallocated space, and the ability to categorize and hash data.

The Windows Registry unallocated space can include deleted records and file fragments. Although complete analysis of the unallocated space is beyond the scope of this thesis, encountered artifacts are included in the findings below. EnCase features the ability to mount various Registry files, including values and records in unallocated space, as entries. To accomplish this, the user must select the two boxes shown in Figure 6.



View File Structure

This file has a "NTRegistry" signature. Continue parsing?

□ Calculate unallocated space

□ Find deleted content

Figure 6. Registry file mount using EnCase.

These entries can be hashed and analyzed to document change as a result of a controlled test. The results generated by a hash analysis alone can be voluminous; fortunately the scope of the results review can be further focused by sorting the dates and times that events occurred. This technique, in addition to corroborating that the Registry recorded events with timeline, was useful in documenting changes in the Registry as a result of the introduction of an external media device.

D. CROSS TOOL VALIDATION

Windows Registry researchers such as Carvey (2011), Lee (2009) and Bunting (2012b) have used the tools discussed above to document mounted device artifacts in the Windows 7 and 8 Registries. Changes within specific hives were noted within previous versions of Windows. These artifacts have a great deal of investigative value when accounting for activity recorded on stand-alone machines. Forensic examiners and incident responders strive to obtain a big picture of the events logged on a given workstation. The observations documented with the set of tools listed above provided a common data set that effectively mapped areas reflecting change as result of mounting an external device.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. FORENSIC METHODOLOGY FOR OBTAINING RELEVANT REGISTRY RECORDS

A. WINDOWS REGISTRY OBSERVATION GOALS

The testing goal was to map the specific Windows 10 Registry values that were modified or created as a result of introducing an external device. It was also important to note the significance of each value change. Items of significance include the associated time and date values, the logged make and model values for each device that was introduced and associated serial number or unique identifiers. The locations of these specific value changes with additional identified artifacts described in the results section were documented to provide a comprehensive map of USB artifacts in the Windows 10 registry.

The workstation used for testing was a Hewlett Packard EliteBook 8570p laptop. This device will be referred to as the Master Control Workstation (MCW) throughout this document. The MCW system specifications are as follows:

- 3rd Generation Intel Core i5-3320M (2.6 GHz, 3 MB L3 cache, 2 cores)
- 16384 MB SODIMMs in both slots (Dual-core processor)
- 2 USB 3.0 ports
- 1394a port
- Optical drive
- eSATA/USB 2.0 combo port
- 750 GB hard drive

B. SYSTEM SET UP CONSIDERATIONS

The test environment was prepared in accordance with best practices established by the National Institute of Standards and Technology (NIST), the International Organization for Standardization (ISO) and the International Association of Computer Investigative Specialists (IACIS) (International Association of Computer Investigative Specialists [IACIS], 2015a). Some of the recommended procedures for the creation of a forensically sound environment include sterilization of the MCW hard drive prior to use, accountability for all data introduced into the MCW hard drive, and write protection of the original items in order to prevent the possibility of unintended modifications of the resident files. Several steps are discussed in the following sections that were taken in the course of this work to yield reproducible results. The obligation to produce results that bear these qualities is highlighted by the NIST testing methodology guidelines and the ISO 5725 document on the accuracy of measurement methods and results. In a court of law, the Daubert (Luttgens & Pepe, 2014) standard applies to the extent that it requires the presentation of forensically sound evidence.

1. Forensic Wipe Requirements

Wiping digital media is a standard forensic practice to ensure that all data that exists on a device is controlled. The practice is generally implemented to ensure that data is not comingled. A drive wipe can be defined as the process of overwriting digital media with known characters (usually 0's) in order to ensure the elimination of all data that may have previously existed on that device. NIST refers to this prepared state of media as "forensically clean." The procedure will also eliminate areas of the hard drive that are normally not accessible by the user such as the manufacturer-installed drive configuration overlay (DCO) and host protected area (HPA) (IACIS, 2015a).

There are UNIX commands, Linux commands, and several free or commercial utilities that can be used to accomplish this task. EnCase Enterprise was used to wipe the hard drive within the MCW prior to the installation of the Windows 10 operating system. At the completion of the wipe procedure, EnCase provided a log that documented that the procedure was successful. A visual inspection was also conducted of the hard drive at the byte level to confirm the success of the wipe operation.

2. Windows 10 Master Copy Installation

The "Windows 10 Technical Preview 9879 (x64) – DVD (English)" was downloaded from the Microsoft Developer Network (MSDN) via the https://msdn.microsoft.com/en-us/subscriptions/keys/ site using a separate computer from the MCW. I burned the ISO file to a DVD and then installed Windows 10 from the DVD to the MCW.

An Administrator account was created within Windows 10 on the MCW. The MCW was then shut down using the power off option within the Windows 10 operating system.

3. EnCase Evidence File

Generally a forensic image will contain a bit-for-bit record of the original evidence item. The E01 image format includes some checksums to ensure that the integrity of the forensic image is preserved. EnCase breaks up the original evidence image into blocks that are 2048 MB in size by default. This block size can be easily modified by the forensic examiner. Each of the chunks is assigned a cyclical redundancy check (CRC) value in the course of creating the image. The CRC values are comprised of 16 or 32 bit values and assist in ensuring the integrity of the individual data blocks when acquiring a device with EnCase (Bunting, 2012a). At the end of the image creation process, a 128-bit Message Digest 5 (MD5) value is generated to ensure the integrity of the E01 evidence file. An abstract view of the EnCase evidence file is shown in Figure 7:

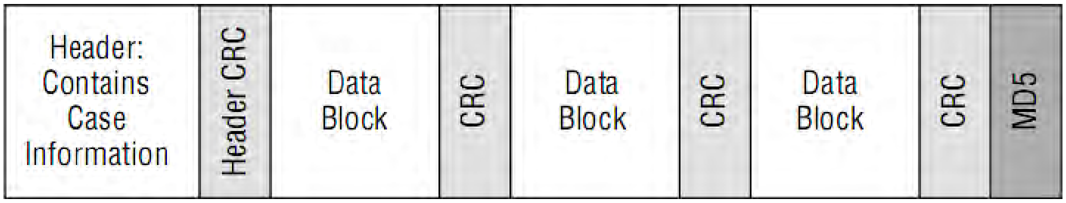


Figure 7. EnCase evidence file structure (from Bunting, 2012a)

The E01 image file format can be logically copied to other storage devices for archiving or further forensic analysis. This image file format can be verified using EnCase Enterprise or free E01 file viewing applications such as AccessData’s FTK Imager. The verification value generated should always match the value of the original MD5 hash.

4. Master Copy Forensic Image Creation

The hard drive was removed from the MCW after the initial Windows 10 installation and was then connected to a Tableau firmware write block device[1] to eliminate the possibility of changes to the data. EnCase Enterprise v7 was used on a separate computer to create an E01 evidence file of the write blocked MCW hard drive that would serve as the master copy image for testing Windows Registry changes that occur as a result of introducing a test thumb drive. No external devices were connected to the MCW prior to the creation of the master copy image.

5. Evidence File Restoration and External Device Introduction Observations

The E01 master copy image was used to restore the original image to the MCW hard drive for each test conducted with the introduction of a new thumb drive. Restoring a hard drive from the E01 file creates a sector-to-sector clone of the original hard drive to a hard drive indicated by the examiner (Bunting, 2012a). The hard drive selected for the restoration has to be of equal or larger size than that of the device that the image was created from. The EnCase option to overwrite remaining sectors after the allocated files were written to the test hard drive was selected.

Following the image file restoration, the MCW was powered on and the time associated with this action was noted. The times associated with system login and thumb drive introduction to the system were also logged for future comparison with the records interpreted with the Registry analysis tools used in this work. Two thumb drives were introduced to the MCW to gather the Windows Registry artifact results summarized in this document. The selected devices were a SanDisk Extreme SDCZ80-032G 32 GB thumb drive bearing 8M140924838B and a Kingston DT101 G2 16 GB thumb drive. The thumb drives were connected to the MCW laptop for twenty minutes and then removed using the “Eject Mass Storage” option from the Windows operating system.

[1] Tableau write block devices are commercially available hardware that block write operations to a connected storage device.

C. REGISTRY FILE REVIEW

The methodology utilized in my review of the registry hives was split into two categories: live and postmortem. Changes logged within the registry hives were actively monitored using ProcMon. RegRipper, EnCase, and RegShot were used to passively observe and validate the overlapping results observed from the respective registry hives, specifically the NTUser.dat, Software hive and System hive. These are hives where USB artifacts were located in previous versions of the Registry (IACIS, 2015b).

1. ProcMon Version 3.10 Active Monitoring

For this method of Windows Registry review, the SysInternalsSuite.zip file was copied to the MCW using a Windows drag and drop from a CD that was introduced. The Procmon.exe file was launched from the Desktop of the MCW and automatically began to display a detailed list of running processes. The event capture process was temporarily paused and the capture options were then modified so that only Windows Registry-related values that wrote to the registry would be captured. Narrowing the focus of the capture events to these operations facilitated the documentation of Windows Registry values that were created or modified as a direct result of introducing the two test thumb drives described above. The capture process was then resumed immediately before introducing one of the test thumb drives to the MCW. The Registry values that were displayed by ProcMon were visually monitored and documented for a 20 minute cycle. No further operations were executed on the MCW during this period. At the end of this interval, I removed the test thumb drive using the "Safely Remove Hardware and Eject Media" feature within Windows. The ProcMon capture feature was then paused and the captured events were exported and preserved for a future cross comparison of results.

2. EnCase Enterprise Observation Methodology

After each test thumb drive was introduced to the Windows 10 operating system for a twenty minute interval, the MCW was shut down and the physical hard drive was removed. The hard drive was then attached to a Tableau forensic write block device and a logical image (L01) file was created using EnCase that contained the "config" folder from the "Windows\System32\" file directory and the NTUser.dat file from the

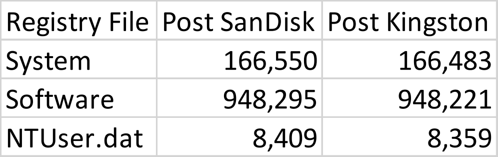
"Users\Jason." The hard drive would then be forensically wiped using EnCase and the controlled master image was restored to the hard drive so that the process could be repeated with the next test thumb drive. It was important to restore the master image to the MCW hard drive so that only results generated by the mounting process of each individual thumb drive could be recorded within the Windows Registry hives.

The Software hive, System hive and NTUser.dat are the traditional locations where USB relevant artifacts have been identified in previous Windows versions (Lee, 2009). These files were mounted within EnCase in order to review their file structure. The allocated and unallocated entries were parsed using the View File Structure feature of EnCase.

The volume of the records from these registry files was substantial. EnCase recovered the following number of subkeys, values and deleted content records from the master image Windows Registry hives that were created immediately after the installation of the Windows 10 operating system and prior to the introduction of either of the test drives:

• NTUser.dat:	8,309
• Software:	946,509
• System:	168,880

The observed quantity of subkeys, values and deleted content records after the introduction of the SanDisk and Kingston test thumb drives to the MCW are listed in Table 1:



Registry File	Post SanDisk	Post Kingston
System	166,550	166,483
Software	948,295	948,221
NTUser.dat	8,409	8,359

Table 1. EnCase Registry observed values

As shown in the above figures, the number of subkeys, values and deleted content records displayed was different for each Registry hive observed and the volume of records stored in each was diverse.

The hash analysis capability of EnCase was used to identify the Registry values that changed as a result of the introduction of the test thumb drives to the MCW. A hash set was created of the System hive, Software hive, and NTUser.dat after the introduction of each test thumb drive. These hash sets were then compared to the respective MCW Registry hives. Differences that were observed in the hash analysis were not solely from the introduction of the test thumb drives. Thousands of changes can occur within the Windows Registry just by the action of powering on MCW. The hash matches were filtered and excluded from review and the resulting data set was then sorted by date and time to identify meaningful results. The hash matches were excluded from review so that only values representing a change after the introduction of a test thumb drive would be displayed.

3. RegShot and RegRipper Data Collection Methodology

The RegShot tool was used to take a snapshot of the Registry hives from the MCW that were exported prior to the introduction of any test thumb drive and compared to the snapshot created of the Registry hives post introduction. RegShot does not allow the user to select individual Registry hives for comparison. The user instead has to specify a directory containing the hive files. Text file and HTML file reports were generated using RegShot. Separate reports were generated to gather results that documented changes generated in the Registry for each test thumb drive introduced. These reports were then searched using artifacts identified in the course of the EnCase review. Table 1 details a comparison of the results identified using this tool.

The RegRipper NTUser.dat, System hive and Software hive plugins were run on these respective registry hives that were extracted after the introduction of each of the test thumb drives. Reports were created to document the results of the plugin parsing of these hives. The Excel spreadsheets generated were reviewed and overlapping data located during the ProcMon and EnCase review were compared and organized into the tables listed in Chapter V of this document.

V. IDENTIFIED USB ARTIFACTS WITHIN THE WINDOWS 10 REGISTRY

This thesis set out with the purpose of answering the question of how it can be forensically determined that a specific USB storage device was connected at some point to a computer system running the Windows 10 operating system (OS). The sections discuss methods that were successful in identifying the affected Registry entries and the shortcomings of other methods. We did not consider Registry changes that were not linked to a known USB identifier. The number of Registry values that are modified when a computer is powered on are voluminous. Registry changes are continuous without any user interaction while the computer is powered on and it is difficult to specifically link a Registry change to a USB device if there are no USB identifiers in a record so we ignored such cases.

A. PROCMON FINDINGS

The location of useful data regarding USB artifacts, just as with any kind of sleuthing, requires a good starting point. The best evidence available is often the kind that makes a record of an event as it occurs. This section will document the initial steps taken in documenting identifying information for the test thumb drives used in this thesis.

1. USB Artifacts Identified

The test thumb drives were initially connected to a forensic workstation configured with the Windows 8.1 Pro N OS using a write blocked device to document the properties and contents of the devices. A 32 GB SanDisk Extreme thumb drive and a 16 GB Kingston DT101 thumb drive were used as test thumb drives for this thesis. The device contents and properties are described in Table 2.

	16 GB Kingston DT101	32 GB SanDisk Extreme
File System	HFS	FAT32
# Files and Folders	722	8091
Used Space/Free Space	4.88 GB/9.64 GB	5.73 GB/24.0 GB
Device Content	Yosemite OSX installation	Windows 8 Installation

Table 2. Test thumb drive contents and properties

In these experiments, ProcMon provided the starting point in the process of identifying unique information specific to the test thumb drives. ProcMon provides real-time monitoring capability of a specified grouping of processes that occur within the Windows OS. A filter was applied that facilitated the viewing of changes occurring exclusively in the Registry. They were introduced to a Hewlett Packard (HP) laptop configured with the Windows 10 OS technical build preview to test how the Registry logged changes when each of the devices were connected. This HP laptop is referred to as the Master Control Workstation (MCW) in this thesis.

The test thumb drives were introduced to the MCW with ProcMon configured to display Registry changes. ProcMon provided a globally unique identifier (GUID) and serial number specific to each test thumb drive immediately after the test USB devices were connected to the MCW as shown in Table 2. GUIDs are also known as universally unique identifiers (UUIDs) and are designed to be unique 128 bit values. The GUID is used to identify a component object model (COM) object within the Registry (Mueller, 2002). A GUID is generated in the Registry when a USB device is connected to a Windows OS. The GUID is a value that is designated by the OS, whereas the serial number is encoded into the USB device. A USB device may have a different GUID assigned if it is attached to a different computer, however the serial number value remains consistent. The serial number and GUID values populated Registry areas identified previously in the Windows 7 OS (Lee, 2009). Experimentation was conducted with attaching the same USB device to the MCW and other test workstations multiple times while running the ProcMon utility. ProcMon showed that the same GUID was associated with the same USB device each time it was connected to the same computer. When the specific USB device was connected to a separate computer running Windows, it was assigned a different GUID. The GUIDs and serial numbers associated with the USB devices provided search strings or keywords to locate other directories within the Registry where these values were stored. ProcMon was also able to log the make and model of the test thumb drives as shown in Table 3.

Identifier	16 GB Kingston DT101	32 GB SanDisk Extreme
Product ID (PID)	6545	5580
Vendor ID (VID)	0930	0781
Serial Number	00D0C9CCDEFCEC50B0006D2A	AA010930142143130243
GUID	f60e3c6d-c4f4-11e4-96e6-6c3be5f58e6f	f60e3e91-c4f4-11e4-96e6-6c3be5f58e6f

Table 3. Identifying artifacts specific to each of the test thumb drives

These values were derived from the ProcMon logged paths listed in Table 4.

Registry Key Path	Forensic Evidentiary Values
HKLM\System\CurrentControlSet\Enum\USB\VID_0781&PID_5580\AA010930142143130243\Properties\{3464f7a4-2444-40b1-980a-e0903cb6d912}\000A\(Default)	VID; PID; Serial Number; First time device was connected after last reboot
HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\{f60e3e91-c4f4-11e4-96e6-6c3be5f58e6f}	GUID

Table 4. ProcMon identified locations displaying data specific to the SanDisk test thumb drive

The VID and PID values listed in Tables 3 and 4 are useful to notate in conjunction with the other identifiers listed above. These are 32 bit values that are hard coded into USB devices by the manufacturer. These values can be researched to identify the make and model of a thumb drive. Performing a text string search of the Registry using only these search strings is a poor option for identifying Registry artifacts, as they are short and the search results include values not necessarily related to a USB device.

The ProcMon findings were exported to Microsoft Excel spreadsheets for further review. Searches were conducted within the exported spreadsheets to locate Registry directories containing the make/model, serial numbers and GUIDs of the test USB devices. The values immediately identified by ProcMon after the test thumb drive introductions to the MCW are documented in Table 5. The same categories of values (make/model, serial number and GUID) were found under the same Registry key path for both thumb drives.

Registry Key Path	Forensic Evidentiary Values
HKLM\System\CurrentControlSet\Enum\USB\VID_0781&PID_5580\AA010930142143130243\Properties\{3464f7a4-2444-40b1-980a-e0903cb6d912}\000A\(Default)	VID; PID; Serial Number; First time device was connected after last reboot
HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\{f60e3e91-c4f4-11e4-96e6-6c3be5f58e6f}	GUID

Table 5. ProcMon identified directories displaying data specific to the SanDisk Extreme device

A search was conducted of the Excel spreadsheet that contained the results of the ProcMon monitoring using search strings that included the identified USB serial numbers and USB GUIDs corresponding to the test thumb drives entered into the Windows find feature. The directories identified as a result of this search are listed in Table 6.

Registry location	Values located
HKLM\SOFTWARE\MICROSOFT\WBEM\WDM\DREDGE \USBSTOR	Make/model, serial number
HKLM\SOFTWARE\MICROSOFT\WINDOWSNT\CURRENTVERSION\EMDMgmt\	Make/model, serial number
HKLM\SOFTWARE\MICROSOFT\WindowsPortable Devices\Devices\	Make/model, serial number
HKLM\System\CurrentControlSet\Enum\STORAGE\VOLUME\	Make/model, serial number
HKLM\System\CurrentControlSet\Enum\SWD\WPDBUSENUM\	Make/model, serial number
HKLM\System\CurrentControlSet\Enum\USB\	VID, PID, serial number
HKLM\System\CurrentControlSet\Enum\USBSTOR\	Make/model, serial number

Table 6. ProcMon identified paths with USB identifiers

2. Registry Changes Documented Upon USB Removal

Prior to removing the test thumb drives from the MCW, ProcMon was used to capture Registry changes that occur when USB devices are removed. In order to accomplish this, the local time and date values were confirmed to be accurate on the MCW. ProcMon was run on the MCW and a test thumb drive was introduced to the MCW. After a five-minute period, the test thumb drive was removed from the MCW using the "Safely Remove Hardware and Eject Media" option. The specific date and time of this option selection and the removal time of the test thumb drive were noted. ProcMon logging was stopped after the removal of the test thumb drive. The ProcMon results were exported to Microsoft Excel spreadsheets for review. This process of ProcMon logging was repeated with the next test thumb drive. Based on the results listed in the Excel spreadsheets corresponding with the removal times of the test devices, a common set of Registry areas were noted for both devices where changes occurred as shown in Table 7.

Procmon Common Results for USB Removal
HKLM\SOFTWARE\MICROSOFT\Windows Search\VolumeInfoCache\C:\DriveType
HKLM\SOFTWARE\MICROSOFT\Windows Search\VolumeInfoCache\C:\VolumeLabel
HKCU\Software\Microsoft\Windows\CurrentVersion\Applets\SysTray\Services
HKCU\Control Panel\Desktop\TranscodedImageCount
HKCU\Software\Classes\Local Settings\MuiCache\1\52C64B7E\LanguageList
HKLM\SOFTWARE\MICROSOFT\WcmSvc\wifinetworkmanager\config\Power
HKCU\Software\Microsoft\Internet Explorer\LowRegistry\Audio\PolicyConfig\PropertyStore\6ec111d2_0\{219ED5A0-9CBF-4F3A-B927-37C9E5C5F14F}\5
HKCU\Software\Microsoft\Internet Explorer\LowRegistry\Audio\PolicyConfig\PropertyStore\6ec111d2_0\{219ED5A0-9CBF-4F3A-B927-37C9E5C5F14F}\4
HKCU\Software\Microsoft\Internet Explorer\LowRegistry\Audio\PolicyConfig\PropertyStore\6ec111d2_0\{219ED5A0-9CBF-4F3A-B927-37C9E5C5F14F}\3
HKCU\Software\Microsoft\Windows\CurrentVersion\ImmersiveShell\StateStore\ItemsStateStoreLastWrite

Table 7. Thumb drive removal changes in Registry

While other changes were noted within ProcMon when the devices were removed, the results are not listed here because no commonality in the processes was noted between the test devices.

A forensic image was created of the MCW config folder and NTUser.dat file after the two test thumb drives were introduced. This forensic image was then processed within EnCase and the common paths listed in Table 5 were reviewed for forensic artifacts. The HKLM\SOFTWARE\MICROSOFT\Windows Search\VolumeInfoCache path had a record for the volume name associated with the last test thumb drive introduced to the MCW. The date and time that was associated with this value did not correspond to the removal time, however the date was accurate when compared to the date the test thumb drives were introduced. It was unclear what time the value in this field corresponded with. The additional paths in Table 6 were also reviewed, however it was not possible to tie any of the times stored in the paths to the date the test thumb drive was actually removed, or other USB identifiers used for searching in this thesis.

B. REGSHOT AND REGRIPPER RESULTS

EnCase was used to access the forensic images created of the MCW Registries. One of these images was created prior to the insertion of either test thumb drive. There were also two separate forensic images of the MCW Registries that were created after each test thumb drive insertion and removal. Prior to the introduction of each test thumb drive to the MCW the original forensic image was restored to the MCW hard drive using EnCase so that the Registry artifacts related to each test thumb drive would not be comingled.

EnCase was then used to export the config folders from each of the images for testing with RegShot. The config folder for the MCW Registry that was created prior to the introduction of either of the test thumb drive was exported to a folder on a forensic workstation labeled “Clean Registry Files.” The config folders for the Registries after the introduction of each of the test thumb drives were exported to folders labeled “SanDisk Registry Files” and “Kingston Registry Files” respectively.

RegShot was initially used to create a snapshot of the config folder within the "Clean Registry Files" folder. RegShot does not have an option for the user to conduct an individual snapshot of the Software, NTUser.dat and System Registry files for exclusive comparison. Instead, a directory containing the config folder contents must be selected. RegShot was then used to create a snapshot from the "SanDisk Registry Files" folder. The "Compare" feature within RegShot was then used to compare the snapshots. This process was repeated to analyze the config files within the "Kingston Registry Files" folder.

The text files generated from the RegShot comparative analyses included "Folders Added," "Folders deleted," "Files added," and "Values added" sections. These sections show values that are drawn only from the Registry entries. The snapshots that were compared of the Registry before and after the introduction of the test thumb drives to the MCW resulted in the Registry changes documented in Table 8.

RegShot Modifications Reported	SanDisk Extreme	Kingston DT101
Folders were added	40	0
Values modified	20	16
Values added	197	3
Files added	3	0

Table 8. RegShot changes noted.

None of the changes recorded appeared to document Windows 10 Registry modifications from a USB device. The RegShot generated text file was searched using text strings that included the test thumb drive make and models, serial numbers and GUIDs and no positive results were observed.

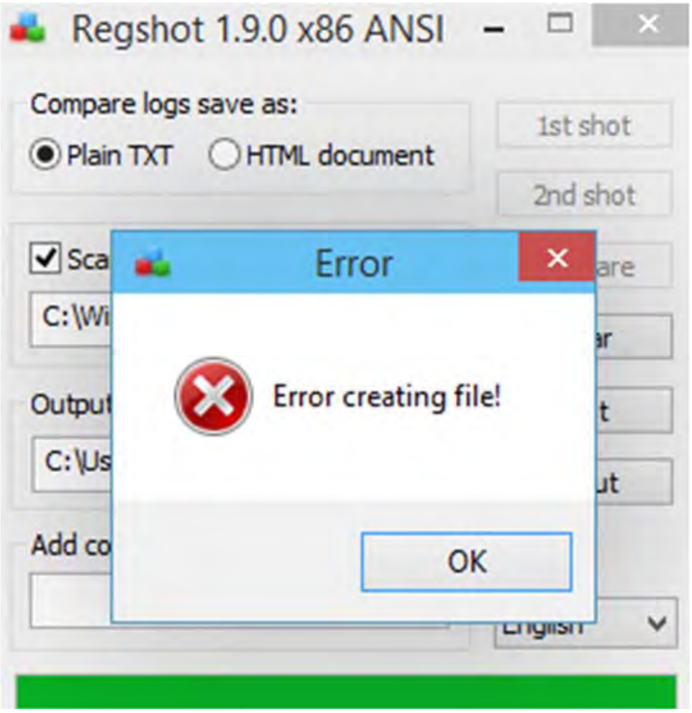
The lack of USB artifact results was unusual and efforts were made to ensure the RegShot application was properly run. RegShot was run on a separate computer running the Windows 8.1 Pro N OS. An initial snapshot of the Windows 8.1 workstation Registry was created prior to the insertion of a thumb drive. A 4 GB PNY thumb drive was then connected to this computer and was left connected to it for five minutes. The "Safely

Remove Hardware and Eject Media" option was then selected within Windows and the thumb drive was removed. The secondary snapshot of the config folder was then created. Comparison with the first snapshot allowed retrieval of information about the PNY thumb drive information, shown in Table 9. This was the only record that confirmed RegShot retrieved USB artifacts for the PNY and this was confirmed via a text string search of the output file using the make/model and serial number as search strings.

Value Added	USB Artifacts
HKLM\SOFTWARE\Microsoft\Wbem\WDM\DREDGE\USBSTOR\Disk&Ven_PNY &Prod_USB_2.0_FD&Rev_0.00\UT16297200003AB4&0_0-{05901221-D566-11d1-B2F0-00A0C9062910}: "LowDateTime:803713417,HighDateTime:0***Binary mof compiled successfully"	Device Make & serial number

Table 9. RegShot USB findings on Windows 8.1 Pro N OS

RegShot was also run on the MCW running the Windows 10 Technical preview OS. Snapshots were created of the Registry before and after thumb drive insertion. The compare utility was then run. The first and second snapshots completed successfully, however the compare snapshot failed and results were not displayed. Several attempts were made to execute the RegShot compare utility on the MCW. These attempts were not successful and resulted in the error listed in Figure 8.


Regshot 1.9.0 x86 ANSI

Compare logs save as:

● Plain TXT ○ HTML document

Error

Error creating file!

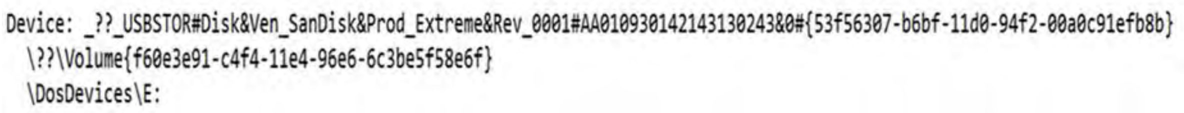
Figure 8. RegShot failed to run on the MCW configured with Windows 10

The RegRipper Software, System and NTUser plugins were run on the Registry files extracted from the MCW after the test thumb drives were introduced. After the plugins were run, a text file was generated detailing the results. A review conducted of the RegRipper output files revealed the list of directories shown in Table 10. These directories were observed for both of the test thumb drives in the RegRipper output files.

Path	Value Contained
SYSTEM\ControlSet001\Enum\USB\	VID/PID and USB serial number
SYSTEM\ControlSet001\Enum\STORAGE\Volume\	make/model/USB serial number
SYSTEM\ControlSet001\Enum\SWD\WPDBUSENUM\	make/model/USB serial number
SYSTEM\ControlSet001\Control\DeviceClasses\	make/model/USB serial number

Table 10. Directories containing USB artifacts as detected using RegRipper

RegRipper identified the volume letter associated with the mounted USB devices, accurately identified the date and time the devices were introduced to the MCW, and the make/model and serial number of the device. The serial numbers were verified to correspond with the test thumb drive serial numbers obtained with the ProcMon utility and also with EnCase as discussed in later sections of this thesis. An example of the RegRipper documentation of the SanDisk test thumb drives shown in Figure 9.



```
Device: _??_USBSTOR#Disk&Ven_SanDisk&Prod_Extreme&Rev_0001#AA010930142143130243&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}
  \??\Volume{f60e3e91-c4f4-11e4-96e6-6c3be5f58e6f}
  \DosDevices\E:
```

Figure 9. RegRipper detection SanDisk USB device

The paths identified by RegRipper in Table 10 were also identified by ProcMon with the exception of SYSTEM\ControlSet001\Control\DeviceClasses\ path. None of the directories identified with RegRipper were identified using the RegShot tool with either of the test thumb drives.

C. ENCASE ANALYSIS RESULTS

1. EnCase Inspection of ProcMon and RegRipper Identified Directories:

Two forensic images were created of the config folder and NTUser.dat files for analysis with EnCase. One of the images was created after the Kingston test thumb drive was introduced to the MCW and the second was created after the SanDisk thumb drive was presented.

EnCase was initially used to navigate to the Registry paths identified using the ProcMon and RegRipper results documented in the above listed tables. The dates and times documenting the test thumb drive times of insertion were found to be accurate to the minute within EnCase. The paths were verified within the Registry files specific to both of the test USB devices and the common findings with the artifacts observed using EnCase are listed in Table 11.

HKLM\System\CurrentControlSet\Enum\USB\	Date/time thumb drive attached/PID/VID
HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\	GUID; Date/time thumb drive attached
HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\CPC	Date/time thumb drive volume dismounted
SYSTEM\ControlSet001\Enum\STORAGE\Volume\	Date/time thumb drive attached/serial number/ make/model
SYSTEM\ControlSet001\Enum\SWD\WPDBUSENUM\	Date/time thumb drive attached/ make/model/ serial number
SYSTEM\ControlSet001\Control\DeviceClasses\	GUID; Date/time thumb drive attached
HKLM\SOFTWARE\MICROSOFT\WBEM\WDM\DREDGE\USBSTOR\	Make/model/serial number/ date & time attached
HKLM\SOFTWARE\MICROSOFT\WINDOWS NT\CURRENTVERSION\EMDMgmt\	Make/model/serial number/ date & time attached
HKLM\SOFTWARE\MICROSOFT\Windows Portable Devices\Devices\	Make/model/serial number/ date & time attached

Table 11. EnCase validation of data previously located with RegRipper and ProcMon

After reviewing the results in Table 11, the Registry changes identified for test thumb drive removal were revisited to review if changes were documented in path HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\ when the thumb drives were removed. The comprehensive ProcMon records did not show changes made to this path when the test USB devices were removed. There were records in the ProcMon spreadsheets indicating modification to these paths shortly after the test thumb drives were introduced to the MCW. This path showed accurate removal dates and times for both of the test thumb drives used in this thesis.

2. Directories Identified through EnCase Indexed Searches

Encase has the capability to mount Registry keys and parse the allocated and unallocated space. The allocated and unallocated spaces were indexed using EnCase. Indexed searches were then conducted of the allocated and unallocated space using the search strings DT101, Kingston, SanDisk, serial number AA010930142143130243 and serial number: 00D0C9CCDEFCEC50B0006D2A. As a result of these searches, additional directories were identified containing USB Registry artifacts for both of the test thumb drives are listed in Table 12:

Windows\System32\config\RegBack\SYSTEM	Make/model/serial number
Windows\System32\config\RegBack\SOFTWARE	Make/model/serial number
SYSTEM\SYSTEM\ControlSet001\Control\DeviceContainer \SWD\WPDBUSENUM\	Make/model/serial number
SYSTEM\ControlSet001\Enum\SWD\PRINTENUM\	Make/model/serial number
SYSTEM\ControlSet001\Hardware Profiles\UnitedVideo\CONTROL\VIDEO\	Make/model/serial number

Table 12. Directories with USB artifacts as identified using indexed searches

EnCase text string searches were also conducted with a selection of file names obtained from the contents of the test thumb drives. A sample of the Kingston and SanDisk test thumb drive files and folders is listed in Table 13.

Kingston	SanDisk
Install OS X Yosemite.app	$WinPEDriver$
coreservices	.Spotlight-V100
iacorestorage	Bootcamp
pkginfo	Boot
SharedSupport	BootCamp

Table 13. Test thumb drive text search strings based on content.

Searches conducted with the terms listed in Table 13 did not result in the identification of Registry artifact that appeared specifically associated with either of the test thumb drives. Any resulting values were inspected for indications of date and time matches and none were apparent.

3. EnCase Hash Analysis Results Justify the Need for Index Searches

The methodology used in creating the hash sets involved mounting the Registry keys within EnCase and parsing the records of the allocated and unallocated space. The Software, System and NTUser.dat keys were each mounted and the contents of these keys were hashed using the MD5 algorithm. The hash sets were observed to have multiple duplicate MD5 values. A hash set was created from the MD5 hash values associated with the Software, System and NTUser.dat that were extracted from the MCW prior to the introduction of either test thumb drive. This hash set will be referred to as the *control hash set*. EnCase was used to create separate hash sets of the Software, System and NTUser.dat files that were extracted after each of the respective test thumb drives were introduced to the MCW. The control hash set was run against the keys extracted after the test thumb drives were presented to the MCW. This process was run with the intention of omitting a large set of data from the EnCase review. Within EnCase, a filter can be applied to omit items that match a hash set from review. While the process was successful in decreasing the number of Registry values for manual review by thousands of entries, it was not successful in uncovering directories containing USB device-specific information. Many hash values in the Registry have nothing to do with values that change as a result of the introduction of a USB device to a computer. This particular approach was not a feasible method for isolating comprehensive Registry values generated from USB devices.

4. Indexed Searches and the Identification of Deleted Values

Indexed searches for known values aid in identifying Registry directories of interest. EnCase detected over 80,000 values that were deleted from the System Registry hive after the SanDisk test thumb drive was introduced to the MCW. The significance of this finding is that while ProcMon, RegRipper and RegShot can detect some Registry-affected areas in the allocated space, they cannot account for a bigger picture of unallocated Registry directories and values. The unallocated search capability of EnCase made text string searches valuable in our experiments.

The values listed in Table 14 were known to be in unallocated space because they were listed as deleted entries with Registry paths in EnCase. After the EnCase searches were conducted, a filter was applied so that only directories and values with a unique hash value were displayed. The results were sorted by item path in Table 14 to identify deleted directories containing the search values.

CsiTool-CreateHive-{00000000-0000-0000-0000-000000000000}\ControlSet001\Control\DeviceClasses\{f33fdc04-d1ac-4e8e-9a30-19bbd4b108ae}\##?#SWD#WPDBUSENUM#_??_USBSTOR#Disk&Ven_SanDisk&Prod_Extreme&Rev_0001#AA010930142143130243&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}#{f33fdc04-d1ac-4e8e-9a30-19bbd4b108ae}
CsiTool-CreateHive-{00000000-0000-0000-0000-000000000000}\ControlSet001\Enum\STORAGE\Volume_??_USBSTOR#Disk&Ven_SanDisk&Prod_Extreme&Rev_0001#AA010930142143130243&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}
CsiTool-CreateHive-{00000000-0000-0000-0000-000000000000}\ControlSet001\Enum\SWD\WPDBUSENUM_??_USBSTOR#Disk&Ven_SanDisk&Prod_Extreme&Rev_0001#AA010930142143130243&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}
CsiTool-CreateHive-{00000000-0000-0000-0000-000000000000}\ControlSet001\Hardware Profiles\UnitedVideo\CONTROL\VIDEO\{F93C235A-A387-49F3-AB88-9B7747BEA741}\0000\##?#SWD#WPDBUSENUM#_??_USBSTOR#Disk&Ven_SanDisk&Prod_Extreme&Rev_0001#AA010930142143130243&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}#{6ac27878-a6fa-4155-ba85-f98f491d4f33}

Table 14. Sample of deleted Registry directories containing USB specific data

Thousands of values and directories were identified in the unallocated space of the Registry files reviewed using EnCase. Many documents have been written detailing unallocated-space artifacts related to the introduction of various known variables within the Registry (Thomassen, 2008). The directories identified in the above descriptions are a sample of forensic artifacts recovered in the testing for this document.

D. SUMMARY OF RESULTS

The tools evaluated in this research had varying degrees of success in identifying USB artifacts that are beneficial to a CFE. The live monitoring capability of ProcMon was helpful in obtaining an initial listing of Registry areas demonstrating change as a direct result of test thumb drive presentation. RegShot testing was unsuccessful and it is possible that there was a compatibility issue with this program and the Windows 10 OS, especially because the program was successfully run on a Windows 8 test system. The RegRipper plugins were successful in retrieving USB artifacts from directories they were written to scan. EnCase was the most useful tool for validating the findings of other tools and providing the ability to review allocated and unallocated records. Table 15 displays the Registry paths that were identified in the course of this research and the tools that were able to note USB artifacts in these locations.

Registry Path with USB Forensic Artifact	EnC	Re	Re	Pr
HKLM\System\CurrentControlSet\Enum\USB\VID_0781&PID_5580\AA010930142143130243\Properties\{3464f7a4-2444-40b1-980a-e0903cb6d912}\000A\(Default)	X			X
HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\{f60e3e91-c4f4-11e4-96e6-6c3be5f58e6f}	X			X
HKLM\SOFTWARE\MICROSOFT\WBEM\WDM\DREDGE\USBSTOR	X			X
HKLM\SOFTWARE\MICROSOFT\WINDOWSNT\CURRENTVERSION\EMDMgmt\	X			X
HKLM\SOFTWARE\MICROSOFT\WindowsPortable Devices\Devices\	X			X
HKLM\System\CurrentControlSet\Enum\STORAGE\VOLUME\	X		X	X
HKLM\System\CurrentControlSet\Enum\SWD\WPDBUSENUM\	X		X	X
HKLM\System\CurrentControlSet\Enum\USB\	X		X	X
HKLM\System\CurrentControlSet\Enum\USBSTOR\	X			X
SYSTEM\ControlSet001\Control\DeviceClasses\	X		X	
SYSTEM\ControlSet001\Hardware Profiles\UnitedVideo\CONTROL\VIDEO\	X			
(Unallocated)SYSTEM\ControlSet001\Control\DeviceClasses\	X			
(Unallocated)HKLM\System\CurrentControlSet\Enum\STORAGE\VOLUME\	X			
(Unallocated)HKLM\System\CurrentControlSet\Enum\USBSTOR\	X			
(Unallocated)SYSTEM\ControlSet001\Hardware Profiles\UnitedVideo\CONTROL\VIDEO\	X			

Table 15. Complete listing of paths holding test USB drive specific artifacts

VI. CONCLUSIONS AND FUTURE WORK

A. IMPACT

This document provides a starting point for Computer Forensic Examiners (CFEs) tasked with analyzing the Windows 10 Registry. This will be important in the near future as Windows 10 will be officially released on July 29, 2015, and it will be the default OS installed on many brands of newly purchased computers. Taking a cue from Mac OS, Windows 10 will also be available as a free upgrade for Windows 7 and Windows 8 users. As discussed in Chapter II, the majority of users (58%) preferred Windows 7 and approximately 15% of users had desktops configured with the Windows 8 OS (Netmarketshare, 2015). Windows 10 marketing is advertising that this OS will combine the strengths of Windows 7 and 8 into a familiar interface and it is likely that a large percentage of users will accept the free upgrade to Windows 10 (Microsoft, 2015). This implies that there will be a global increase in the number of Windows 10 systems analyzed as a result of criminal and internal investigations.

The Registry documents many modification and write operations and this contributes to the overall stability of the OS. This information provides a wealth of forensic artifacts for a CFE. Some of the paths identified in this document identify the date and time a thumb drive of specific make, model and serial number was introduced to or removed from a Windows 10 OS. This information is useful in identifying user activity and monitoring when specific actions occurred on a computer. It can be especially helpful in malware investigations because thumb drives are an important source of malware.

B. FUTURE WORK

The artifacts presented in this thesis are only a start regarding forensically useful Windows 10 Registry artifacts, as only USB-related artifacts were discussed. For example, the Windows Registry can also document changes that occurred as a result of installing and running a program, the uniform resource locator (URLs) entries entered into an Internet browser by a specific user, and the Registry changes that document the option to bypass the recycle bin when deleting files so that they go directly to unallocated space. Research will have to be conducted to tie an artifact with an action and tying an action with an activity executed by a user. Windows 10 also has many new features that will offer new Registry artifacts of interest.

Microsoft Edge, a new Internet browser, allows the user to create notes related to websites visited through the browser and to share pages with others. It will likely provide new kinds of artifacts. Cortana is a personal assistant with a notebook feature capable of tracking high volumes of user information such as interests and favorite places (Microsoft, 2015). If the feature is enabled, it will no doubt hold useful information for a CFE. Windows 10 will also be compatible with certain cellular phones, tablets and computers, and the behavior across these devices may show new artifacts.

C. CONCLUDING REMARK

Windows 10 contains forensically uncharted features and the Registry artifacts related to these features will require time to identify. The Registry in previous versions of Windows has contained many useful artifacts for CFEs that show indicators of user behavior and assist in the documentation of events that occurred on a system. Registry information can supplement an interview while a search warrant is taking place, or provide answers for various system actions when few other sources are available.

LIST OF REFERENCES

Bunting, S. (2012a). *EnCase computer forensics: The official enCase certified examiner study guide* (3rd ed.). Indianapolis, IN: John Wiley and Sons.

Bunting, S. (2012b). *Mastering windows network forensics and investigation* (2nd ed.). Indianapolis, IN: John Wiley & Sons.

Carvey, H. (2011). *Windows registry forensics: Advanced digital forensic analysis of the windows registry*. Burlington, MA: Syngress.

International Association of Computer Investigative Specialists (IACIS). (2015a). *Ethics & standards*. In Basic computer forensic examiner (BCFE) manual (pp. 356-361). Retrieved from https://www.dropbox.com/sh/vyylqshrl7vbt1r/AADJdOD zj3FVuEsR0KjEKHUja/IACIS%202015%20BCFE%20MANUAL%20FOR%20 PRINTER?dl=0

International Association of Computer Investigative Specialists (IACIS). (2015b). *Windows registry.* In Basic computer forensic examiner (BCFE) manual (pp. 555-571). Retrieved from https://www.dropbox.com/sh/vyylqshrl7vbt1r/AADJdOD zj3FVuEsR0KjEKHUja/IACIS%202015%20BCFE%20MANUAL%20FOR%20 PRINTER?dl=0

Internet Crime and Complaint Center (IC3). (2008). *IC3 2008 annual report on internet crime released*. Retrieved from https://www.ic3.gov/media/2009/090331.aspx

International Organization for Standardization (ISO). (1994). *Accuracy of measurement methods and results*. Retrieved from https://www.iso.org/obp/ui/#iso:std: iso:5725:-1:ed-1:v1:en

Lee, R. (2009). *USBKEY-Guide* [PDF document]. Retrieved from https://blogs. sans.org/computer-forensics/files/2009/09/USBKEY-Guide.pdf

Luttgens, J., & Pepe M. (2014). *Incident response & computer forensics* (3rd ed.). New York City, NY: McGraw-Hill.

Microsoft. (2015). *Windows 10: It's the windows you know only better*. Retrieved from http://www.microsoft.com/en-us/windows/features

Mueller, J. (2002). *Special edition using soap*. Indianapolis, IN: Que Publishing.

National Institute of Standards and Technology (NIST). (2001). *General test methodology for computer forensic tools* [WORD document]. Retrieved from www.cftt.nist.gov/Test%20Methodology%207.doc

Netmarketshare. (2014). *Desktop operating system market share*. Retrieved from https://www.netmarketshare.com/operating-system-market-share.aspx?qprid=10&qpcustomd=0

Smith, F. & Bace, R. (2003). *A guide to forensic testimony: The art and practice of presenting testimony as an expert technical witness*. Boston, MA: Pearson Education, Inc..

Statistic Brain Research Institute. (2014). *Average cost of hard drive storage*. Retrieved from http://www.statisticbrain.com/ average-cost-of-hard-drive-storage/

Thomassen, J. (2008). *Forensic analysis of unallocated space in windows registry hive files* (master's dissertation). Retrieved from http://sentinelchicken.com/data/ JolantaThomassenDISSERTATION.pdf

Windows Dev Center. (n.d.). *Symbolic links*. Retrieved May 9, 2015, from https://msdn.microsoft.com/en-us/library/windows/desktop/aa365680%28v-vs.85%29.aspx

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California